Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method in a computer system for serializing hardware reset requests in a software communication request queue in a processor card, said processor card processing software requests utilizing said queue in a serial order, said computer system including said processor card and a second resource card, said method comprising the steps of:

receiving, within said processor card ~~from an application~~, a hardware reset request that requests said processor card to reset said second resource card;

placing said hardware reset request in said queue that is included in said processor card;

processing requests from said queue in said serial order, said hardware reset request being processed when all requests from said queue currently being serviced have completed being serviced; and

resetting, by said processor card, said second resource card in response to said hardware reset request being processed.

2. (Previously presented) The method according to claim 1, further comprising the steps of:

said computer system including a plurality of resource cards, said second resource card included within said plurality of resource cards;

said processor card coupled to said plurality of resource cards utilizing a single reset bus; and

resetting, by said processor card utilizing said reset line, all of said plurality of resource cards simultaneously in response to said receipt of said hardware reset request.

3. (Previously Presented) The method according to claim 2, further comprising the steps of:

receiving software communication requests and hardware reset requests;

placing said software communication requests and hardware reset requests in said queue in said serial order in which said software communication requests and hardware reset requests were received, said hardware reset requests being serialized within said queue with said software communication requests.

4. (Original) The method according to claim 2, further comprising the steps of:

looking at a next request in said serial order in said queue;

determining whether said next request is a hardware reset request;

in response to a determination that said next request is a hardware reset request, determining whether all of said plurality of resource cards have completed servicing of any pending software communication requests; and

waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending software communication requests.

5. (Previously presented) The method according to claim 4, further comprising the steps of:

in response to determining that all of said plurality of resource cards have completed servicing of any pending software communication requests, determining whether all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests; and

waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests.

6. (Previously presented) The method according to claim 1, further comprising the steps of:

said processor card coupled to said second resource card utilizing a hardware reset line; and

resetting, by said processor card, said second resource card by pulling said hardware reset line high.

7. (Previously presented) The method according to claim 1, further comprising the steps of:

said second resource card including a microcontroller;

said hardware reset request requesting said processor card to reset said microcontroller; and

resetting, by said processor card, said microcontroller in said second resource card in response to said hardware reset request being processed.

8. (Original) The method according to claim 2, further comprising the steps of:

each one of said plurality of resource cards including a microcontroller, a memory, and synchronization bits;

utilizing said synchronization bits to maintain information about current servicing of software communication requests by each one of said plurality of resource cards;

resetting said microcontroller and said synchronization bits in each one of said plurality of resource cards simultaneously in response to said receipt said hardware reset request specifying one of a plurality of resource cards to reset; and

losing said information about current servicing of software communication requests by each one of said plurality of resource cards when said synchronization bits are reset.

9. (Currently amended) A computer system for serializing hardware reset requests in a software communication request queue in a processor card, said processor card processing software requests utilizing said queue in a serial order, said computer system including said processor card and a second resource card, said system comprising:

said processor card receiving~~, from an application,~~ a hardware reset request that requests said processor card to reset said second resource card;

said queue, which is included in said processor card, for storing said hardware reset request;

said queue for processing requests that are stored in said queue in said serial order, said hardware reset request being processed from said queue in said serial order when all requests from said queue currently being serviced have completed being serviced; and

said processor card resetting said second resource card in response to said hardware reset request being processed.

10. (Previously presented) The system according to claim 9, further comprising:

said computer system including a plurality of resource cards, said second resource card included within said plurality of resource cards;

said processor card coupled to ~~of~~ said plurality of resource cards utilizing a single reset bus; and

said processor card ~~for~~ using said reset line to reset all of said plurality of resource cards simultaneously in response to said receipt of said hardware reset request.

11. (Previously presented) The system according to claim 10, further comprising:

said queue for storing software communication requests and hardware reset requests in said serial order in which said software communication requests and hardware reset requests were received, said hardware reset requests being serialized within said queue with said software communication requests.

12. (Original) The system according to claim 10, further comprising:

said system including a CPU executing code for looking at a next request in said serial order in said queue;

said CPU executing code for determining whether said next request is a hardware reset request;

in response to a determination that said next request is a hardware reset request, said CPU executing code for determining whether all of said plurality of resource cards have completed servicing of any pending software communication requests; and

said processor card for waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending software communication requests.

13. (Previously presented) The system according to claim 12, further comprising:

in response to determining that all of said plurality of resource cards have completed servicing of any pending software communication requests, said CPU executing code for determining whether all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests; and

said processor card for waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests.

14. (Previously presented) The system according to claim 9, further comprising:

said processor card coupled to said second resource card utilizing a hardware reset line; and

said processor card resetting said second resource card by pulling said hardware reset line high.

15. (Previously presented) The system according to claim 9, further comprising:

said second resource card including a microcontroller;

said hardware reset request requesting said processor card to reset said microcontroller; and

said processor card resetting said microcontroller in said second resource card in response to said hardware reset request being processed.

16. (Original) The system according to claim 10, further comprising:

each one of said plurality of resource cards including a microcontroller, a memory, and synchronization bits;

said synchronization bits for maintaining information about current servicing of software communication requests by each one of said plurality of resource cards;

said microcontroller and said synchronization bits being reset in each one of said plurality of resource cards simultaneously in response to said receipt said hardware reset request specifying one of a plurality of resource cards to reset; and

said information about current servicing of software communication requests being lost by each one of said plurality of resource cards when said synchronization bits are reset.

17. (Currently amended) A computer program product that is stored in a storage medium in a computer system for serializing hardware reset requests in a software communication request queue in a processor card, said processor card processing software requests utilizing said queue in a serial order, said computer system including said processor card and a second resource card, said product comprising:

first instruction means for receiving, within said processor card ~~from an application~~, a hardware reset request that requests said processor card to reset said second resource card;

second instruction means for placing said hardware reset request in said queue that is included in said processor card;

third instruction means for processing requests from said queue in said serial order, said hardware reset request being processed when all requests from said queue currently being serviced have completed being serviced; and

fourth instruction means for resetting, by said processor card, said second resource card in response to said hardware reset request being processed.

18. (Previously presented) The product according to claim 17, further comprising:

said computer system including a plurality of resource cards, said second resource card included within said plurality of resource cards;

said processor card coupled to said plurality of resource cards utilizing a single reset bus; and

fifth instruction means for resetting, by said processor card utilizing said reset line, all of said plurality of resource cards simultaneously in response to said receipt of said hardware reset request.

19. (Previously presented) The product according to claim 18, further comprising:

sixth instruction means for receiving software communication requests and hardware reset requests;

seventh instruction means for placing said software communication requests and hardware reset requests in said queue in said serial order in which said software communication requests and hardware reset requests were received, said hardware reset requests being serialized within said queue with said software communication requests.

20. (Previously presented) The product according to claim 18, further comprising:

sixth instruction means for looking at a next request in said serial order in said queue;

seventh instruction means for determining whether said next request is a hardware reset request;

in response to a determination that said next request is a hardware reset request, eighth instruction means for determining whether all of said plurality of resource cards have completed servicing of any pending software communication requests; and

ninth instruction means for waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending software communication requests.

21. (Previously presented) The product according to claim 20, further comprising:

in response to determining that all of said plurality of resource cards have completed servicing of any pending software communication requests, tenth instruction means for determining whether all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests; and

eleventh instruction means for waiting to execute said hardware reset request until all of said plurality of resource cards have completed servicing of any pending sequence of multiple software communication requests.

22. (Previously presented) The product according to claim 17, further comprising:

said processor card coupled to said second resource card utilizing a hardware reset line; and

fifth instruction means for resetting, by said processor card, said second resource card by pulling said hardware reset line high.

23. (Previously presented) The product according to claim 17, further comprising:

said second resource card including a microcontroller;

said hardware reset request requesting said processor card to reset said microcontroller; and

fifth instruction means for resetting, by said processor card, said microcontroller in said second resource card in response to said hardware reset request being processed.

24. (Previously presented) The product according to claim 18, further comprising:

each one of said plurality of resource cards including a microcontroller, a memory, and synchronization bits;

sixth instruction means for utilizing said synchronization bits to maintain information about current servicing of software communication requests by each one of said plurality of resource cards;

seventh instruction means for resetting said microcontroller and said synchronization bits in each one of said plurality of resource cards simultaneously in response to said receipt said hardware reset request specifying one of a plurality of resource cards to reset; and

eighth instruction means for losing said information about current servicing of software communication requests by each one of said plurality of resource cards when said synchronization bits are reset.